A NATURE INSPIRED FIREFLY HEURISTIC APPROACH ON OPTIMIZATION OF MULTI OBJECTIVE JOB SHOP SCHEDULING PROBLEMS

1K.C.UDAiyAKumar, 2M.CHANDRASEKARAN

1Research Scholar, Sathyabama University, Chennai, Tamilnadu, India.
2Director, Department of Mechanical Engineering, Vel’s University, Chennai, Tamilnadu, India
E-mail: kcudaiyakumar1967@gmail.com

Abstract- Due to rapid increase in modernization and globalization in today’s economy, every manufacturing industry faces great challenges to minimize their manufacturing cost and to sustain in the competitive market. Scheduling is the earmarking of resources over time to effectuate an agglomeration of tasks ordained in any field of engineering and non engineering. Job shop scheduling problem is the most labyrinthine and paradigmatic problem of all kinds of production scheduling problems. Preponderance of JSSP is buttoned down into non deterministic (NP) hard problem because of its complexity and entanglement. Scheduling are generally elucidated by using heuristics to obtain optimal or near optimal solutions because problems found in practical applications cannot be fathomed to optimality using available resources in many cases. Many researchers attempted to iron out the problem by applying various optimization techniques. While using traditional methods they observed huge impasse in deciphering high complex problems and meta-heuristic algorithms were proved most efficacious algorithms to solve various JSSP so far. The objective of this paper i) to bestow a newly developed meta heuristic called Firefly algorithm (FA) because of inspiration on Firefly and its characteristic. ii) To treasure out the combined objective function by determining optimal make span, mean flow time and tardiness of different size problems using two Bagchi Job Shop Scheduling Problems (JSP-I and JSP – 2) and iii) to prove that the proposed FFA algorithm is a gratifying problem solving technique for JSSP with multi criteria.

Keywords- Benchmark, Fire-fly, Job Shop Scheduling Problem, Make Span, Mean Flow Time, Tardiness.

I. INTRODUCTION

Job shop scheduling problem is one of the most gargantuan combinatorial problems Scheduling is the allocation of resources over time to percolate a collection of tasks .The job shop scheduling problem(JSP) is embodied by a set m machines {M_1,M_2,......M_m}, and a collection of n jobs {J_1,J_2,......J_n} to be scheduled ,where each job must pass through each machine once only.Each job has its own processing order and this may bear no relation to the processing order of the any other job. Job Shop Scheduling problems are NP-hard problem, so its ramifications is more.

Various optimization approaches have been widely applied to solve the JSSP. Conventional methods based on mathematical methods and /or full numerical search (for example, Branch and Bound artificial intelligence and Lagrangian Relaxation can guarantee the optimal solution. They have been propitiously used to solve the JSSP.

However, these methods highly monopolize computational time and resources even for solving moderately-large problem size and therefore chimerical if the computational limitation exists. Later, a larger size JSSP has been construed by an approximation optimization methods or meta-heuristics such as Tabu search and simulated annealing. Two jobs to be performed by three machines: (2×3) job shop scheduling problem is illustrated in Table 1. In this problem, each job requires three operations to be processed on a pre-defined machine sequence. The first job (J_1) needs to be initially operated on machine M_1 for 10 time units and then sequentially processed on M_2 and M_3 for 9 and 8 time units, respectively. Likewise, the second job (J_2) has to be initially performed on M_2, M_2 and for 9, 8, 7time units respectively. The design task for solving JSSP is to rummage through the best schedule(s) for operating all pre-defined jobs in order to optimize either single or multiple scheduling objectives. An example of 2-jobs 3-machines scheduling problem with processing times.

Table 1 –An example of 2- jobs 3- machines scheduling problem with processing times

<table>
<thead>
<tr>
<th>Job</th>
<th>Operation Time (O_k)</th>
<th>Time (T_k)</th>
<th>Machine (M_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>0_13 10</td>
<td>10 M_1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0_12 9</td>
<td>- M_2</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>0_13 8</td>
<td>- M_3</td>
<td>-</td>
</tr>
<tr>
<td>J2</td>
<td>0_23 9</td>
<td>- M_1</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>0_21 8</td>
<td>8 M_2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0_22 7</td>
<td>- M_3</td>
<td>7</td>
</tr>
</tbody>
</table>

A Nature Inspired Firefly Heuristic Approach On Optimization of Multi Objective Job Shop Scheduling Problems
II. FIREFLY ALGORITHM

A. Inspiration and algorithm

Fireflies, belonging to the family of Lampyridae, are tiny winged beetles having the capability of producing light. Firefly algorithm developed by Xin-She Yang (2008), is one of the au courant meta-heuristics. Firefly algorithm rhapsodizes some of the idiosyncrasies of the firefly behavior. They follow three rules: a) all the fireflies are unisex, b) each firefly is attracted only to the fireflies that are brighter than itself; Strength of the attractiveness is proportional to the firefly’s brightness, which attenuates over the distance; the brightest firefly moves randomly and, c) brightness of every firefly determines it’s quality of solution ; in most of the cases, it can be proportional to the objective function. Using the above three rules, a pseudo-code of the Firefly Algorithm may look as follows:

Algorithm 1: Basic Firefly Algorithm Pseudo-Code

Input: f(x,f,t), x = (x₁, x₂, ........., xₜ); //Objective function
n, I₀, □, α;
// User-defined constants
Output: find ∑ (S_min,Fmean,Tmean) // position of

for i ← 1 to n do
xᵢ ← Initial Solution ();
end

While termination requirements are not met do

min ← arg min ( f(xᵢ) );
for i ← 1 to n do
for j ← 1 to n do
if f(xᵢ) < f(xⱼ) then
dᵢⱼ ← Distance(xᵢ,xⱼ); //move xᵢ
towards xⱼ
β ← Attractiveness (I₀, □,dᵢⱼ);
Xᵢ’ ← (1-β) Xᵢ + β Xⱼ + α (Random () - 1/2); // movement
end
end

xₘᵢₙ ← xₘᵢₙ * α (Random () -1/2); // best briefly moves randomly
end

In the above algorithm , n is the number of the fireflies, I₀ is the light intensity at the source, □ is the absorption coefficient and α is the size of the random step . All these parameters will be explained further in detail.

III. APPLICATION OF FIREFLY ALGORITHM FOR JOB SHOP SCHEDULING

A. Introduction

The objective of this paper is i) to capitalize a recently developed meta heuristic called Firefly algorithm (FA) because of inspiration on Firefly and its characteristic. ii) to find the combined objective function on, multi objectives of JSSP (i.e. make span minimization, tardiness and mean flow time) using Bagchi’s two Jobshop problems named JSP1 and JSP2. iii) The analysis of the experimental results on Firefly algorithm is compared with other algorithms with best known solution literature review.

B. Firefly evaluation

The next stage is to measure the flashing light intensity of the firefly, which depends on the problem considered. In this work, the evaluation on the goodness of the schedules is measured by the makespan, which can be calculated using equation (1), where C_k is the completed time of job k. m_i and T_i represents mean flow time and tardiness.

Minimizes C_max = max(C_1,C_2,C_3,.....,C_n) ,m_i and T_i

C. Distance

The distance between any two fireflies i and j at X_i and X_j, respectively, can be defined as Cartesian distance (r_ij) using equation (2), where X_ik is the component of the spatial coordinate x_i of the i^th firefly and d is the number of dimensions.

r_ij = √(x_i-x_j)^2

D. Atractiveness

The calculation of attractiveness function of a firefly is shown in equation (3), where r is the distance between any two fireflies,β₀ is the initial attractiveness r=0, and γ is an absorption coefficient which controls the decrease of the light intensity[1,2].

β₀+β₀exp(-γ r^m), with m≥1

E. Movement

The movement of a firefly i which is attracted by a brighter firefly j is given by the following equation (4), Where x_i is the current position or solution of a firefly, The attractiveness of a firefly by the adjacent fireflies is given by β₀ X exp(-γ r^m) X (x_j – x_i)

The random movement OF Firefly –  α(rand – 1/2)

A Nature Inspired Firefly Heuristic Approach On Optimization of Multi Objective Job Shop Scheduling Problems

72
The coefficient $\alpha$ is a randomization parameter determined by the problem of interest with $\alpha \in [0, 1]$. While $\text{rand}$ is a random number obtained from the uniform distribution in the space $[0, 1]$. 

$$x_i = x_i + \beta_0 \exp(-\gamma r^{2}_{ij}) X (x_i - x_j) + \alpha (\text{rand} - \frac{1}{2})$$

(4)

Experiments
A. Multi-objective context
Multi objective optimization differs from single objective optimization in many ways. For two or more conflicting objectives, each objective corresponds to different optimal solutions but none of these trades off solutions is optimal with respect to all objectives. Hence multi objective optimization does not try to figure out one optimal solution but optimizes all trade off solutions.

Apart from having multi objectives, the fundamental difference is multi objective optimization deals with two goals, the first goal is to hunt for a set of solutions as close as possible to pareto- optimal front and the second goal is to find a set of solutions as diverse as possible. This work using two JSP given by Bagchi are the basis of the following experiments. The first problem is called JSP-1, is a ten job five machine instance and the second problem is called JSP-2, is ten job and ten machine instance. Performance measures are make span, mean flow time and mean tardiness. Make span ($ms_i$) is defined as maximum completion time of all jobs. Mean flow time ($mf_i$) is the average of the flow times of all jobs. Mean tardiness ($T_i$) is defined as the average of tardiness of all jobs.

The combined objective function for the multi objective Job shop problem is

$$\text{COF}=\min\{w_1(ms_i/ms*)+w_2(T_i/T*)+w_3(mf_i/mf*)\}$$

Where, $w_1=(R1/R), w_2=(R2/R), w_3=(R3/R), R=(R1+R2+R3),$ Where $R1,R2,R3$- Random numbers

$ms*$ Make span Global minimum
$T*$ Mean tardiness Global minimum
$mf*$ Mean flow Time Global minimum
$T_i$ Mean tardiness Iteration minimum
$mf_i$ Mean flow Time Iteration minimum

w1, w2, w3 Weightage factors

COF combined objective Function

These recently developed algorithms have been applied by few researchers for solving optimization problems in various engineering field. In this work, the settings of FFA parameter such as number of fireflies ($n$), number of generations/iterations ($G$), the light absorption coefficient ($\gamma$), randomization parameter ($\alpha$) and attractiveness value ($\beta$) have to be chosen in an ad hoc fashion. Generally the combination factor (nG) predisposes the amount of search in the solution space conducted by this algorithm.

This factor is directly akin to the size of the problem considered. In this research, the acceptable computational limitations are practically materialized, therefore the combination factor was fixed at 1000 in order to accommodate computational search within the time limit. The light absorption coefficient ($\gamma$) was varied from 0 to 10, the randomized parameter was usually set between 0 to 1 and the attractiveness function was also chosen between 0 to 1.

B. Experimental Design and Analysis
In order to solve the optimization problem, Matlab under Windows XP operating system have to be necessarily accomplished. The following parameter used in solving JSSP are $\alpha = 0.05, \beta_0 = 0.02, \gamma =0.0001$, random number (step size) $m=1$ which are being obtained by sensitivity analysis keeping number of fireflies as 10 and maximum generation of fireflies as 100, hence total no of functional evolution is 1000.

The results of computational experiments for Bagchi 2 Job shop problems are shown in Table 2 & 3 and Fig. 1& 2. These results are compared with genetic algorithm results experimented by previous researcher as shown in Table 2 & 3.

<table>
<thead>
<tr>
<th>Solution</th>
<th>FFA</th>
<th>Genetical Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>156</td>
<td>10.8</td>
</tr>
<tr>
<td>2</td>
<td>160</td>
<td>15.7</td>
</tr>
<tr>
<td>3</td>
<td>160</td>
<td>14.1</td>
</tr>
<tr>
<td>4</td>
<td>161</td>
<td>11.9</td>
</tr>
<tr>
<td>5</td>
<td>162</td>
<td>16.4</td>
</tr>
<tr>
<td>6</td>
<td>163</td>
<td>15.6</td>
</tr>
<tr>
<td>7</td>
<td>165</td>
<td>13.7</td>
</tr>
<tr>
<td>8</td>
<td>167</td>
<td>15.1</td>
</tr>
<tr>
<td>9</td>
<td>169</td>
<td>16.1</td>
</tr>
<tr>
<td>10</td>
<td>182</td>
<td>15.8</td>
</tr>
</tbody>
</table>

Table 2-Comparisons of JSP 1 Pareto solution of FFA with GA

<table>
<thead>
<tr>
<th>Solution</th>
<th>FFA</th>
<th>Genetical Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>156</td>
<td>10.8</td>
</tr>
<tr>
<td>2</td>
<td>160</td>
<td>15.7</td>
</tr>
<tr>
<td>3</td>
<td>160</td>
<td>14.1</td>
</tr>
<tr>
<td>4</td>
<td>161</td>
<td>11.9</td>
</tr>
<tr>
<td>5</td>
<td>162</td>
<td>16.4</td>
</tr>
<tr>
<td>6</td>
<td>163</td>
<td>15.6</td>
</tr>
<tr>
<td>7</td>
<td>165</td>
<td>13.7</td>
</tr>
<tr>
<td>8</td>
<td>167</td>
<td>15.1</td>
</tr>
<tr>
<td>9</td>
<td>169</td>
<td>16.1</td>
</tr>
<tr>
<td>10</td>
<td>182</td>
<td>15.8</td>
</tr>
</tbody>
</table>

Table 3- Comparisons of JSP 2 Pareto solution of FFA with GA
The recently developed firefly algorithm is the most puissant and stalwart technique used to solve the problems of Job Shop Scheduling in Optimized way. It is one of the intelligible methods which are ease in applying on any NP hard problem. This can be interpreted proudly as this algorithm was exercised to find the minimization of makespan \((C_{\text{max}})\), mean flow time and tardiness using two Bagchi JSP problem. The parameters of FFA such as the absorption coefficient, the population of fireflies and the number of iterations depends upon the optimized problem.

The experimental results clearly shows that the proposed approach is able to find a solution of set of solutions close to the pareto optimal front as well as find a set of diverse solution. Combined objective values of above both the problems were exactly matching and few of the problems have values lesser than GA. It is quite clear that this algorithm is one of the best heuristic approaches to solve multi objective criteria in JSSP.

**REFERENCES**


